



# Development of a videogame to improve the math skills of children with dyscalculia

Boris Goshev Gerasimov

Final Degree Work  
Bachelor's Degree in  
Videogame Design and Development  
Universitat Jaume I

June 3, 2019

Supervised by: José Ribelles Miguel

# ACKNOWLEDGMENTS

First of all, I would like to thank my Final Degree Work supervisor, José Ribelles Miguel for all the support, kindness and advices for my work.

I also would like to thank Sergio Barrachina Mir and José Vte. Martí Avilés for their inspiring [LaTeX template for writing the Final Degree Work report](#), which I have used as a starting point in writing this report.

# ABSTRACT

The project consists of the development of a didactic videogame that helps to improve the skills of the children with dyscalculia. The videogame is presented as an alternative to traditional math exercises, to make it less boring for the children and that the experience is more enjoyable, attractive and fun.

This will be achieved by adapting the traditional math exercises to mini-games. The difficulty will adapt to the player depending on the failures that makes in each of the mathematical exercises. The game will be available for computers and android tablets. It will be developed with Unity3D.

# CONTENTS

<b>Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>1 Technical Proposal</b>	<b>1</b>
1.1 Introduction and motivation of work . . . . .	1
1.2 Related subjects . . . . .	2
1.3 Objectives of the Bachelor's Thesis . . . . .	2
1.4 Expected results . . . . .	2
1.5 Tools . . . . .	2
1.6 Task planning and timing . . . . .	3
<b>2 Game Design Document</b>	<b>4</b>
2.1 Dyscalculia . . . . .	4
2.1.1 Competitors . . . . .	5
2.2 Overview . . . . .	6
2.2.1 Game concept . . . . .	6
2.2.2 Genre . . . . .	6
2.2.3 Target audience . . . . .	6
2.2.4 Educational objective . . . . .	7
2.3 Gameplay and Mechanics . . . . .	7
2.3.1 Gameplay . . . . .	7
2.3.2 Controls and Mechanics . . . . .	7
2.3.3 Score . . . . .	8
2.4 Levels . . . . .	9
2.5 Narrative design . . . . .	11
2.5.1 Story and narrative . . . . .	11
2.5.2 Characters . . . . .	11
2.6 Interface . . . . .	11
2.6.1 HUD . . . . .	11



2.6.2	Music and sound effects . . . . .	12
2.7	Game Art . . . . .	12
<b>3</b>	<b>Development process</b>	<b>14</b>
3.1	Programming the videogame . . . . .	15
3.1.1	Menu Scene . . . . .	15
3.1.2	Main Game Scene . . . . .	15
3.1.3	Mini-games Scenes . . . . .	16
3.1.3.1	First mini-game . . . . .	16
3.1.3.2	Second mini-game . . . . .	16
3.1.3.3	Third mini-game . . . . .	17
3.1.4	Dialogue System . . . . .	17
3.1.5	Screen resolutions and scale game objects . . . . .	18
3.1.6	Saving and loading system . . . . .	18
3.1.7	Particles System . . . . .	19
3.1.8	Audio Source . . . . .	19
3.2	2D Art and 3D Modeling . . . . .	20
3.2.1	Menu . . . . .	20
3.2.2	3D Game Scene . . . . .	21
3.2.3	Mini-games . . . . .	22
3.2.4	Character . . . . .	24
3.2.5	2D Toys . . . . .	25
3.2.6	UI Elements . . . . .	25
3.2.6.1	Buttons . . . . .	25
3.2.6.2	Fonts . . . . .	26
3.2.6.3	Icon . . . . .	26
3.2.6.4	Textures for particles . . . . .	26
3.3	Music . . . . .	27
3.4	2D/3D Animations . . . . .	27
<b>4</b>	<b>Results</b>	<b>28</b>
4.1	Work development . . . . .	28
4.1.1	Project deviations . . . . .	29
4.1.1.1	Temporary deviations . . . . .	30
4.2	Results . . . . .	30
<b>5</b>	<b>Conclusion</b>	<b>35</b>
5.1	Conclusions . . . . .	35
5.2	Future work . . . . .	36
	<b>Bibliography</b>	<b>37</b>

## LIST OF FIGURES

1	Example outline of the videogame mechanics . . . . .	8
2	Example outline of the first mini-game . . . . .	9
3	Example outline of the second mini-game . . . . .	10
4	Example outline of the third mini-game . . . . .	10
5	Images of Rayman Adventures in the videogame . . . . .	12
6	Rayman Adventures . . . . .	13
7	Images of Pancakes & Milkshakes in the videogame . . . . .	13
8	<i>DialogueTrigger</i> Script component window of Unity . . . . .	18
9	<i>CanvasScaler</i> component window of Unity . . . . .	19
10	<i>Particles System</i> component window of Unity . . . . .	20
11	Background of the menu scene . . . . .	20
12	Models of the environment and trees . . . . .	21
13	First final environment with materials . . . . .	21
14	Second final environment with materials . . . . .	22
15	Third final environment with materials . . . . .	22
16	Background for the first mini-game . . . . .	23
17	Background for the second mini-game . . . . .	23
18	Background for the third mini-game . . . . .	24
19	2D Sprites of Matitas and his states . . . . .	24
20	2D Sprites of all the toys . . . . .	25
21	Menu icons . . . . .	25
22	Buttons and the dialogue platform . . . . .	26
23	Icons for the executable videogame . . . . .	26
24	Texture for the <i>Particles System</i> . . . . .	26
25	Initial menu of Dystoy . . . . .	31
26	Main Game Scene before first mini-game . . . . .	32
27	Main Game Scene before second mini-game . . . . .	32
28	First mini-game scene . . . . .	33
29	Second mini-game scene . . . . .	33
30	Third mini-game scene . . . . .	34

## LIST OF TABLES

1.1	Estimated hours of work . . . . .	3
4.1	Hours of work done compared to the estimated hours . . . . .	29

# TECHNICAL PROPOSAL

## Contents

1.1	Introduction and motivation of work . . . . .	1
1.2	Related subjects . . . . .	2
1.3	Objectives of the Bachelor's Thesis . . . . .	2
1.4	Expected results . . . . .	2
1.5	Tools . . . . .	2
1.6	Task planning and timing . . . . .	3

## 1.1 Introduction and motivation of work

Dyscalculia affects the learning of mathematics in people. This entails great difficulty in daily life and in certain cases can cause school failure. It has no cure but can be improved with constant practice [1]. As we already know, traditional mathematical exercises can be very hard for children, to the point that they decline to solve them. As a student, I have had difficulties with maths. I can't imagine how difficult must be for a child to deal with dyscalculia, already in itself the mathematics entails a difficulty.

Having said that, my proposal is to create a videogame to help these children improve their math skills in a more enjoyable and fun way [2]. It will be a free application in which the priority is to adapt the experience of the player based on his difficulty with mathematics, and of course the fun. I will investigate the most general symptoms of dyscalculia and try to adapt them in the form of mini-games. The player can practice doing a mini-game that is based on a specific symptom. I will also investigate what exercises and in what form they have to be exposed so that the improvement in the learning of the mathematics is achieved.

## 1.2 Related subjects

- VJ1227 - Game engines
- VJ1223 - Videogame art
- VJ1222 - Videogame conceptual design
- VJ1238 - Foundations for educational videogame design

## 1.3 Objectives of the Bachelor's Thesis

The main objectives are the following:

- Achieve a colorful and childish art, that is attractive and pleasant for a young audience.
- Adapt the traditional mathematical exercises based on a specific symptom and turn it into a functional mini-game that helps in the learning of mathematics in that particular aspect.
- Present a functional demo of the videogame with three mini-games.

The secondary objective:

- Getting the player to feel identified with the main character.

## 1.4 Expected results

I hope to get a functional demo of the videogame that can help people with dyscalculia and with practice improve their skills in learning mathematics. Also, the game is composed of several mini-games somehow focus on a symptom. The videogame is aimed at a children's audience, and that the players feel comfortable playing it, and have fun, don't think about the failure and the pressure to do it well, just practice to overcome the difficulty with the mathematics.

## 1.5 Tools

- Unity3D: Game engine that will be used to develop the videogame.
- MonoDevelop: Code editor for programming in C#.
- Adobe Photoshop: Graphics editor for drawing the artistic assets.
- Blender: 3D graphics software tool to create the models.

- LibreOffice: Free office suite to write the technical proposal and the game design document.
- MiKTeX: Text editor to prepare the final memory using LaTeX.

## 1.6 Task planning and timing

	Initial tasks	Estimated hours
<b>1. Documentation</b>	1.1 Technical proposal	5
	1.2 Game design document	10
	1.3 Prepare the final memory	50
	1.4 Prepare the presentation	10
<b>2. Analysis</b>	2.1 Research on dyscalculia	20
<b>3. Design</b>	3.1 Game design	20
	3.2 Art design	50
<b>4. Development</b>	4.1 Programming the videogame	100
	4.1.1 Menu	15
	4.1.2 In Game	25
	4.1.3 Mini-game 1	20
	4.1.4 Mini-game 2	20
	4.1.5 Mini-game 3	20
	4.2 Retouching and detail implementation	25
<b>5. Testing</b>	5.1 Final testing	10
<b>Total:</b>	<b>300</b>	

Table 1.1: Estimated hours of work

# GAME DESIGN DOCUMENT

## Contents

2.1	Dyscalculia . . . . .	<b>4</b>
2.1.1	Competitors . . . . .	5
2.2	Overview . . . . .	<b>6</b>
2.2.1	Game concept . . . . .	6
2.2.2	Genre . . . . .	6
2.2.3	Target audience . . . . .	6
2.2.4	Educational objective . . . . .	7
2.3	Gameplay and Mechanics . . . . .	<b>7</b>
2.3.1	Gameplay . . . . .	7
2.3.2	Controls and Mechanics . . . . .	7
2.3.3	Score . . . . .	8
2.4	Levels . . . . .	<b>9</b>
2.5	Narrative design . . . . .	<b>11</b>
2.5.1	Story and narrative . . . . .	11
2.5.2	Characters . . . . .	11
2.6	Interface . . . . .	<b>11</b>
2.6.1	HUD . . . . .	11
2.6.2	Music and sound effects . . . . .	12
2.7	Game Art . . . . .	<b>12</b>

## 2.1 Dyscalculia

To begin it is necessary to explain what the dyscalculia consists of and how can be treated for it is improvement. Dyscalculia is a specific disorder of learning of neurobiological origin that affects the obtaining of knowledge about numbers and calculation in the framework of the normal intellectual level, and that is not caused by a bad learning

method. This disorder can manifest itself in a very heterogeneous way. The most usual thing is that the person experiences difficulty with the most basic aspects of numerical processing and calculation. For example, understand the meaning of numbers, quantities, difficulty in writing or reading them. On the other hand, in daily life these difficulties can be interpreted, for example to read the time on a clock, handle amounts of money, in the workplace and a long etc. The other problem is that dyscalculia is sometimes accompanied by attention deficit or motor development (in a lesser percentage), as well as emotional or language problems.

There are 5 major types of dyscalculia and each of them has a specific difficulty or similar difficulties. The videogame will focus on the most general exercises that can help in the 5 types. After selecting the exercises, they should be designed in such a way that they are brief, attractive, have easy-to-understand instructions and are adapted to the person's capacity so that the complexity increases gradually. Besides, the exercises can be adapted in the videogame, and the most essential thing is that it is motivating for the child and also needs to have a striking appearance.

An important part of the research is to see the problem from another perspective. When making a didactic videogame it is essential to empathize with the problem or the situation, to put yourself in its place to help. This is achieved by consulting specialized centers, reading articles on dyscalculia and testing applications focused on the problem. One of the things that most helps sympathize with people who suffer from this disorder is the dyscalculia simulator [3]. It is a perfect way to know how these people feel.

Finally it is necessary to investigate existing videogames with the same objective to see their weak and strong points (which will be explained in the next section). So at the time of the development of the game can be taken into account these points to get ideas or even improve their weaknesses.

Before starting the development of the game, the Game Design Document (Chapter 2) has been made. It specifies all the information necessary for the preparation of the project, based on all the information collected together with the competitors (next section). Throughout the development of the project there have been small modifications in some sections, which will be explained later in (Section 4.1.1).

### 2.1.1 Competitors

As mentioned previously, I searched for existing videogames with the same objective as the project. Practically there are no videogames destined to the dyscalculia, only a few. The majority of the videogames are for the cognitive training but they are not really focused in the dyscalculia. This is a problem for people with this disorder because these applications will not help them to better understand numbering and calculations. They have exercises of operations, memory, visual perception, etc. But they are very complicated exercises and do not have a configuration for the difficulty. On the other hand most of these applications have very unattractive interfaces and no motivational aspects for the user.



The two main videogames that are focused on dyscalculia are the following:

- Meister Cody Talasia [4]: It is a pedagogical videogame for children aged to 6-12 years. Talasia has a great variety of mathematical exercises and it is structured by linear levels. Also is quite attractive visually. All levels go by time. There is reaction levels, enumeration of points, comparison of numbers, audio listening of the number with selection, calculations and memory, and about 19 different exercises. This videogame is not free, initially they provide you with 4 free trial days of a single level, is a monthly payment. As well tells a little story about the world of Talasia. It is a good videogame but its access can be difficult for some people due to the economic issue.
- Blips! [5]: This is a videogame focused on dyscalculia. Blips has a small variety of exercises and is not very polished graphically, but in each level it shows a very funny different robot. It does not have a story to follow. On the other hand, the mechanics of the exercises are similar to the previously mentioned videogame. The game is not free but is a single purchase. Economically it is more accessible than the Meister Cody Talasia.

## 2.2 Overview

### 2.2.1 Game concept

The game will be called Dystoy and will be a 3D first person videogame in which you will be presented with mathematical mini-games in 2D. The main character has been lost in the world of toys and needs the help of the player to be able to return to the village of toys. The player has to solve a series of mini-games in order to reach the goal. The mini-games will focus on the difficulties that people with dyscalculia have.

### 2.2.2 Genre

Dystoy is a serious educational videogame since its purpose is to improve the math skills of children with dyscalculia and also to make learning in a more fun way. It is a different form to traditional practical exercises but currently has a very positive effect.

### 2.2.3 Target audience

The target audience are children with dyscalculia. Mainly children who start primary education. When they are introduced to mathematics and begin to perform very simple operations. The age at which the dyscalculia usually manifests is from 6 to 8 years. The videogame will focus on an age of 6 to 9 years. On the other hand the mini-games will be adapted to each player regardless of the age range.

### 2.2.4 Educational objective

As we already know, dyscalculia [6] is a disorder that makes learning difficult in mathematics and it is also equivalent to dyslexia, a little-known disorder. The child has difficulty understanding and performing mathematical calculations.

The objective of the videogame is to provide practical help improving the math skills of children with dyscalculia. To achieve the improvement of learning each mini-game will be adapted to a specific type of dyscalculia and in addition to a general symptom that presents this type of dyscalculia. The most frequent symptoms are the difficulty of recognizing numbers, arithmetic symbols, performing basic operations (addition, subtraction, multiplication and division) and amnesia with numerical information or visual perception.

On the other hand, they have difficulty in consulting the score during the games, for this fact the videogame will not have visual score but it will have an internal score to adapt the difficulty avoiding the anxiety of the player. Players do not have to worry about whether they have correctly performed the calculations. The most important thing is to learn from the exercise performed.

## 2.3 Gameplay and Mechanics

### 2.3.1 Gameplay

It is a very young audience, the videogame has to be concise, clear and manageable, to have a good accessibility for the player. The objective of the videogame is to help the main character to return to the village of toys. To achieve this the player is presented with a number that he must memorize. Each of the digits of the number is a specific route. These routes at the same time are mini-games that the player must solve to select the next route.

Not knowing how to answer or solve an operation generates stress in the player, and this is not the goal. For this reason in Dystoy there are no enemies since the challenges that are presented to the player can take some difficulty.

### 2.3.2 Controls and Mechanics

As it is an educational game the controls aren't complicated, is a point clicker videogame. You can only play with the mouse. At the start of the game, as mentioned before, a random number will be displayed that the player must memorize. When the player is ready he must click on the number. Once clicked, three routes will be shown, each one with a digit (see Figure 1). The player has to click on the same digit as the first digit of the memorized number. This is a way of keeping the player concentrated at all times, and to relate the two digits. The identification of digits generates great difficulty for the people with dyscalculia.

On the screen there will always be a button that if is clicked will show the number to memorize, since for each digit there is a specific mini-game and it can be difficult to remember each digit of the number. This button will only be accessible if the player fails to choose the route. If it fails to select the digit of the route, the digits of the routes will be changed again and the button will be enabled, since it is a sample of difficulty. If the player decides to consult the number constantly doesn't matter because indirectly is relating the digits. Also, at the time of selecting the route there will not be a timer due to the fact that it generates pressure on the player and isn't what we are looking for. The selection of the route will be shown in 3D with the floating digits while the mini-games will be presented in 2D. To finish the videogame will consist of three mini-games that will be explained in the section of levels.

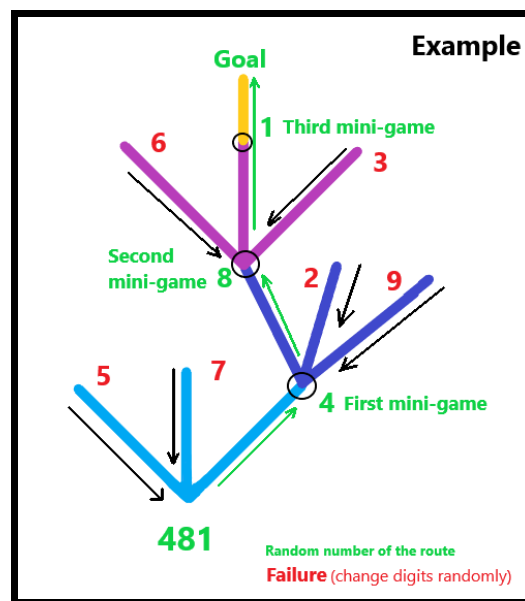


Figure 1: Example outline of the videogame mechanics

### 2.3.3 Score

Numbers and mathematics generate a lot of anxiety for people with dyscalculia. As I mentioned earlier there will not be an external scoring system for the player. But there will be an internal scoring system that allows me to adapt the difficulty for the player. Depending on the number of failures that the player makes, the difficulty will be adapted based on these failures. As well, if the player has more success in the exercises, the difficulty will increase slightly. The difficulty of the exercises will be adapted to the user.

## 2.4 Levels

In the beginning there will be three mini-games and each one of them will focus on a type of dyscalculia. The first mini-game will focus on the ideognostic dyscalculia. It consists of the difficulty to make mental operations, without using the numbers to reach the result, and also to understand concepts or ideas related to mathematics or arithmetic. They also have difficulties remembering mathematical concepts. This mini-game will consist of counting the number of toys that appear on the screen and the player has to choose the correct number (see Figure 2). Counting numbers is a very effective and practical mental exercise.

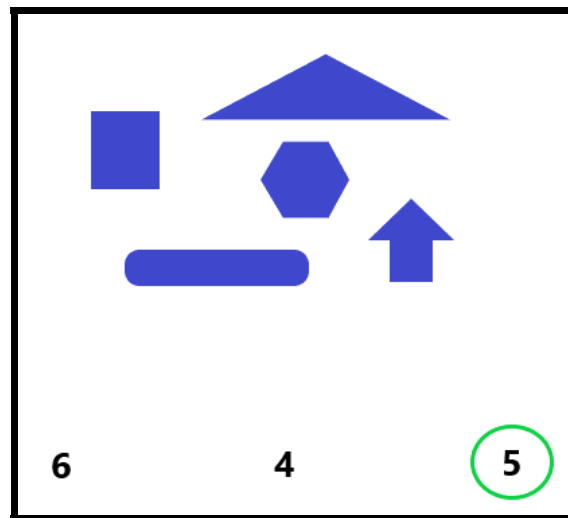


Figure 2: Example outline of the first mini-game

The second mini-game will be based on a combination of two types of dyscalculia, the practognostic, which consists in having difficulties in enumerating, comparing and manipulating numbers. The operational one is the difficulty of manipulating numbers and arithmetic symbols to perform the calculations. This mini-game will consist of resolving operations and then deciding which result is greater or lesser of all (see Figure 3).

In the third mini-game, the player will have aside a different number of toys and on the other side operations (see Figure 4). The player has to relate the number of toys to the result of the operation. This mini-game is a combination of the first mini-game and calculation exercises.

After researching and reading articles, also looking at examples, I have concluded that they are good exercises and that with the constancy and daily training you can get positive effects.

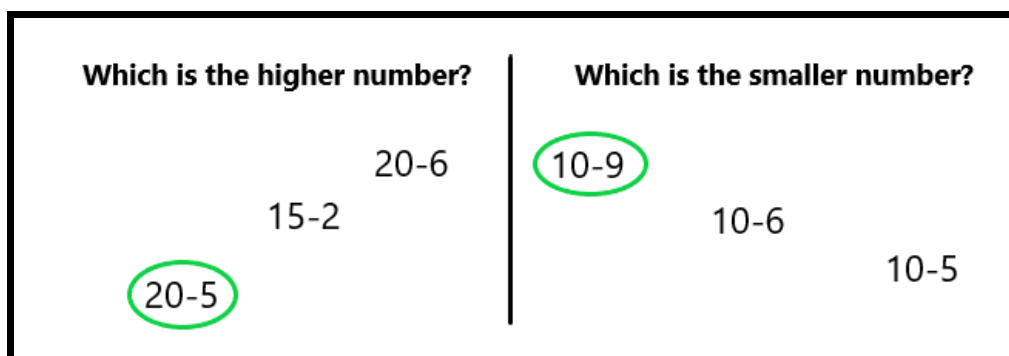


Figure 3: Example outline of the second mini-game

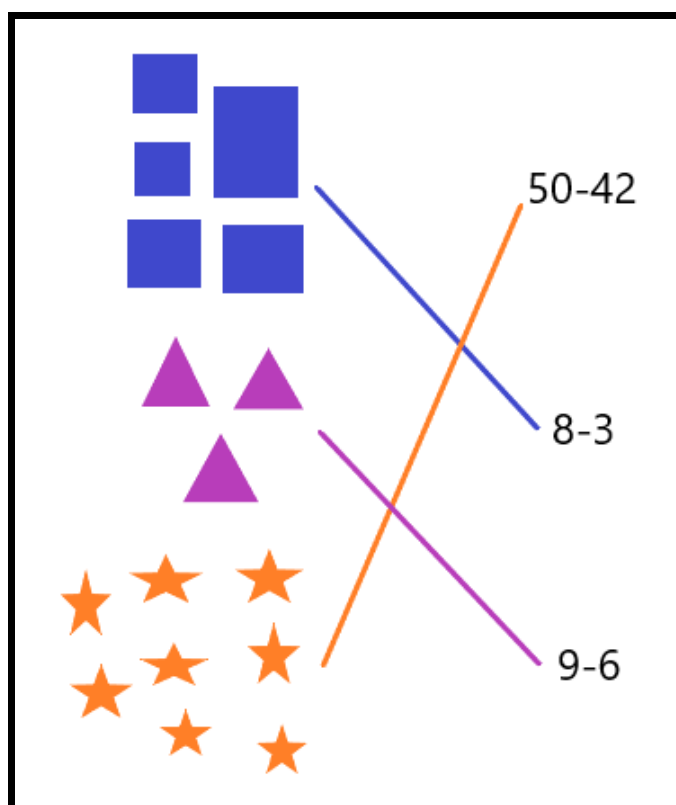


Figure 4: Example outline of the third mini-game

## 2.5 Narrative design

### 2.5.1 Story and narrative

The theme of the videogame will be fantasy. It is a world of toys in which everyone identifies by numbers and each village has a specific number. This happens due to the fact that there are toys that are the same in appearance but different in personality. The main character is called Sam. Apparently one day Sam was playing hide and seek with his friends and he hid in a forest. After a long time Sam realized that he was lost.

After wandering through the forest he found a different being that kept chasing him. Sam decided to stay with him and named him Matitas. He had a big problem he does not remember the way to home, and he is not good with the numbers. All the routes have different digits. What Sam does not know is that Matitas is a mathematical being who knows how to get anywhere at any time, and will help him to get home safe.

### 2.5.2 Characters

Sam is the main character is clueless, friendly and sometimes very shy. He has always had difficulties with mathematics and this made his daily life difficult. And even more in the world of toys where everything was enumerated. Sam will get to understand mathematics better thanks to the help of his partner Matitas.

Matitas is the secondary character who will take care of helping Sam back home. He knows all the ways of the world of toys, since he is very good at mathematics. Matitas was created by mathematical scientists to guide themselves throughout the world of toys. Matitas escaped from the scientists since they only used it as a tool and he looked for a quiet place to live in harmony, until he met Sam and decided to help him. This character is inspired in "Navi" [7] of the "The Legend of Zelda".

Matitas has the appearance of a bird. His own being is a metaphor that he is the guide and that it is associated with wisdom. He will wear a graduated cap that represents intelligence and science. On the other hand, arithmetic symbols are presented in his clothes, to show the relationship with mathematics. The head of Matitas will also have a triangular shape that is interpreted as the Pythagorean Theorem. Finally he will wear a scientific robe for his past in the laboratory.

## 2.6 Interface

### 2.6.1 HUD

Before starting the videogame the main screen will appear where we will find three buttons "Play", "Help" and "Quit". When the game starts in the upper left corner there will always be a button to return to the main menu.

In addition, there will also be a button that will remember the number that the player must memorize, it will only appear if the player fails in the route selection. In

each mini-game you will be previously explained what you have to do to complete it. Also, if the player doesn't clear up in the upper right corner there will be an exclamation button that will show the instructions of the mini-game.

### 2.6.2 Music and sound effects

The background music will be the same for the selection of the route and for the mini-games. It will be an environmental music based on the sound of nature. As the target of the audience is very young the music will be loose and relaxing. Not to stress the player. On the other hand, when the player has to interact with the toys they will be accompanied by effects of funny sounds.

## 2.7 Game Art

At the time of selecting one of the three digits will appear in 3D floating. The environment will have simple 3D models, when the correct digit is selected will be passed to the mini-games with a 2D style. The style of the videogame will be a combination of "Rayman Adventures" [8] (see Figure 5,6) and a fan made videogame called "Pancakes & Milkshakes" [9] (see Figure 7) from the movie "Wreck-It Ralph 2" [10]. Considering the audience has to have very bright colors and a magical environment, which is visually pleasing and very attractive.



Figure 5: Images of Rayman Adventures in the videogame



Figure 6: Rayman Adventures



Figure 7: Images of Pancakes & Milkshakes in the videogame



## DEVELOPMENT PROCESS

## Contents

---

3.1	Programming the videogame . . . . .	<b>15</b>
3.1.1	Menu Scene . . . . .	15
3.1.2	Main Game Scene . . . . .	15
3.1.3	Mini-games Scenes . . . . .	16
3.1.3.1	First mini-game . . . . .	16
3.1.3.2	Second mini-game . . . . .	16
3.1.3.3	Third mini-game . . . . .	17
3.1.4	Dialogue System . . . . .	17
3.1.5	Screen resolutions and scale game objects . . . . .	18
3.1.6	Saving and loading system . . . . .	18
3.1.7	Particles System . . . . .	19
3.1.8	Audio Source . . . . .	19
3.2	2D Art and 3D Modeling . . . . .	<b>20</b>
3.2.1	Menu . . . . .	20
3.2.2	3D Game Scene . . . . .	21
3.2.3	Mini-games . . . . .	22
3.2.4	Character . . . . .	24
3.2.5	2D Toys . . . . .	25
3.2.6	UI Elements . . . . .	25
3.2.6.1	Buttons . . . . .	25
3.2.6.2	Fonts . . . . .	26
3.2.6.3	Icon . . . . .	26
3.2.6.4	Textures for particles . . . . .	26
3.3	Music . . . . .	<b>27</b>
3.4	2D/3D Animations . . . . .	<b>27</b>

---

## 3.1 Programming the videogame

In this section, I will explain how the videogame has been programmed, including all its operations and activities.

The game is made up to 5 scenes. Starting in the menu where you can access the different options. At the beginning of the game you start in the main game scene where you have to complete the objective to unlock the different mini-games. The transition between scenes is performed by the *SceneManager* class that provides Unity, using the *LoadScene("Scene name")* function to load the scene. Next, each of these scenes of how they have been implemented and developed will be explained in each subsection, since the mini-games are independent of the main game scene.

### 3.1.1 Menu Scene

In the game menu there are 4 sections. It can start the game by clicking on *"Play"* button that will redirect you to the main game scene. Also to see the instructions by clicking on *"Help"* button of the main game scene, and the rules of each of the mini-games. Besides, it can access to see the points achieved by pressing *"Points"* button. In this section will appear the total points of success and failures that the player has made during all the time played. On the other hand, it can reset the score from scratch at any time by pressing the *"restart"* button that is located on the upper left side. Finally can exit the videogame by clicking on *"Quit"* to close the application. To manage all the sections Unity UI Buttons have been used.

### 3.1.2 Main Game Scene

First, a random number of 3 non-repeated digits is generated, this number is the route that the player has to follow to unlock the mini-game. The behavior of this scene depends on *FirstManager*, *SecondManager*, *ThirdManager* and *TrueRandomNumber*. Each of these managers is in charge of managing a row of 3 spawn points, in total 9 spawn points. I also make sure that each of the digits of the number selected as a route is in each of the rows that corresponds to the manager. The spawn points invoke game objects that correspond to the 3D model of the digits, these digits are not repeated and will be from 0 to 9. As well are responsible for messing up the positions of the numbers randomly, only if the player fails to select the digit.

The *TrueRandomNumber* script is responsible for generating the random digits and the correct digits that will show with letters. At the beginning of the game the player has to read with letters the digit that is shown in the upper part of the screen and then identify it with one of the three digits that are shown in the central part, and so to continue. Finally, the *ObjectClicker* script found in the *Main Camera* is used to detect the game objects selected by the player. In this way it can detect the "click" that the player makes using the *RaycastHit* class, which has Unity default, so that returns a non-zero value if reaches a collider. Each of the prefabs has a box collider, rigidbody and a

tag that corresponds to the own digit of the 3D model. Using the tag can compare with the digit to know if the player has hit the correct digit or not. When the three digits are correct, a different level is unlocked.

### 3.1.3 Mini-games Scenes

In the following sections, I will explain how has been done and in what briefly consists of the linear levels of the videogame. Being linear does not mean that the first level is easier than the second, each level will have a difficulty that depends on the player. Each of these levels has a different mechanics that is responsible for reinforcing a certain aspect in the learning of mathematics. In addition, will be commented how the difficulty is adjusted automatically to the player. At the end of each level, a button will be displayed that will change to the main game scene in order to unlock the next level.

#### 3.1.3.1 First mini-game

The first mini-game consists of counting the amount of toys that there are, and using the 3 buttons at the bottom of the screen to answer. The behavior of this scene depends on the *SpawnLevel1* script that is part of an rectangular collider, and this collider serves as an area for the toys. The *SpawnObject()* function ensures that toys only appear within the area. It is a way of preventing toys from being created outside the camera and not being visible. On the other hand, the function *PreventSpawnOverlap(Vector3)* serves so that the toys do not overlap one another, since these game objects are 2D images. To work properly, it is necessary to apply colliders to all the 2D images, only if they are going to be used as toys. Also, *SpawnLevel1* script generates the correct number and the random ones that will be shown.

The player will have a score visible in the upper right side of the screen that will show the correct answers. If reaches a certain number of successes, it will go to the next level. The difficulty of the level varies based on the successes and failures of the player, failure does not remove points. Each time the digit is correct, the random range number becomes slightly larger and the correct digit changes, but if it fails, the range of numbers to be displayed is reduced taking into account the correct number without changing until the player succeeds. The buttons update the text they display depending on the values they get from the *SpawnLevel1* script.

In all three levels, I created the *Reseteo()* function to empty two lists and re-generate random numbers. The first list contains the positions of the numbers to determine which button will be displayed. The second list is to avoid repeating the numbers.

#### 3.1.3.2 Second mini-game

The second mini-game is to select the largest digit or the smallest digit using the 4 buttons on the bottom of the screen, this is designed to see if the player can differentiate quantities. The *RandomNumbersLevel2* script is responsible for generating 4 random

numbers mainly in a small range. It can randomly ask, which is the largest number, which is the smallest or repeat the same. All random numbers are in a list so it can get the minimum or maximum depending on the numbers that have touched. Like the previous level, has the correct answers in the upper right part and failing does not remove points. The difficulty has an algorithm similar to the previous section, but with a slight difference. Either if fails or hits the correct number will change. The more points you have, the more the range of random numbers will increase, and the more you fail the more the range of random numbers will be reduced. The range of the numbers will not depend on the correct number. With this modification, the difficulty will increase slightly. The buttons are updated with the values that gets from the *RandomNumbersLevel2* script.

### 3.1.3.3 Third mini-game

In the third mini-game you have to solve the mathematical operation to answer correctly. Only in the case that the player fails the same number of toys as the result will appear, if succeeds, only the operation will appear. All this behavior is handled by the *SpawnArea* script that uses some similar functions of the first level, to avoid overlaps or that the toys go out of the area as discussed in the First mini-game section. On the other hand, a list with the symbols of the addition, subtraction and multiplication is used. It can randomly touch one of these operations depending on the symbol, this also controls the results of the subtractions that are not negative and that the multiplications are simple. If fails, the correct number remains until it success, but the toys that appear on the screen change their position so that they count again, just like the numbers on the buttons. When it success, the difficulty does not change, but if fails, the range of random numbers shown on the buttons is reduced based on the correct number. Generally, the operations are more complicated, so I decided not to increase the difficulty by guessing correctly. When it fails, reduces the difficulty by adding a help element as the toys to be counted.

### 3.1.4 Dialogue System

The dialogue system is composed of three scripts; *DialogueManager*, *DialogueTrigger* and *Dialogue*. The *DialogueManager* is responsible for handling all the sentences. First of all, each of these sentences are glued in a FIFO queue and then only dequeue if the player presses the continue button to move on to the next sentence. Pressing the button checks if has reached the end and there are no more sentences. If otherwise there are more sentences, the function of *StartCoroutine()* (an coroutine) is executed, which receives as an argument another function of type *IEnumerator*, which in turn receives the sentence. The coroutine allows to start a parallel action and then returns an *IEnumerator* object (or nothing). In this case *IEnumerator* will return a letter of the sentence, until completing the entire sentence. When there are no more sentences the animation is activated and the dialogue disappears. The *Dialogue* class is a serializable system, in other words is a struct of the dialog, so that each dialog has the same size in the text area. Finally, the *DialogueTrigger* (see Figure 8) is responsible for activating

the *DialogueManager* to start the dialogue. This method helps when writing dialogs since is not necessary to open the script to modify it, with access to the component of Unity is enough. It can also put the number of sentences and the number of dialogues that you prefer.

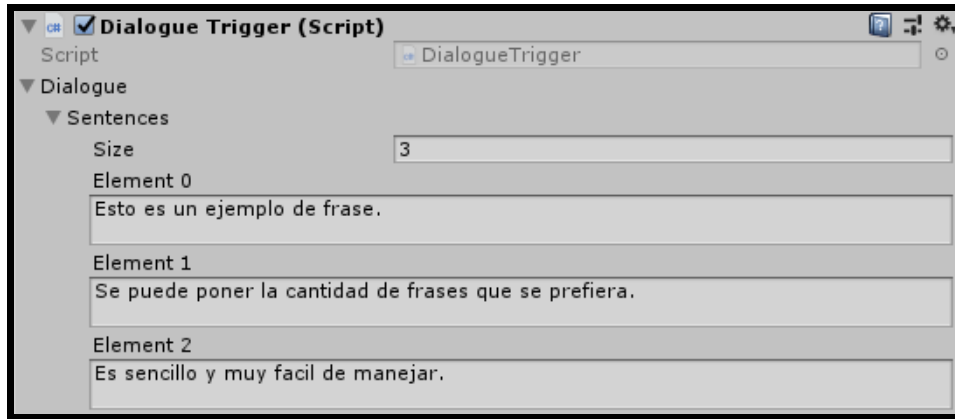


Figure 8: *DialogueTrigger* Script component window of Unity

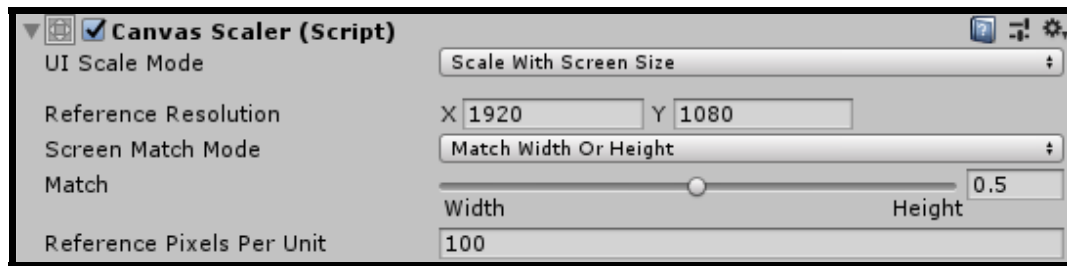
### 3.1.5 Screen resolutions and scale game objects

The game has to be adapted to the different resolutions, since each device has a different resolution. It is very important that all the videogame components are displayed correctly. For this the *CanvasScaler* component has been used (see Figure 9) that allows to control the scale and the density of the pixels of all the elements that are part of the *Canvas*, in this case the HUD. It has been selected the scale mode UI to "*Scale with screen size*", to scale all the elements of the canvas depending on the resolution. Thus all the positions and sizes can be specified according to the pixels of a specific reference resolution, which in this case will be 1920x1080. Even if the resolution of the screen is greater than the reference resolution it will be extended to adapt to the screen and if it is smaller it will be reduced. Also, if a non-uniform scale is presented, the "*Screen Match Mode*" setting in "*Match Width Or Height*" is used so that the axes are not scaled individually, so that the scale is uniform.

*CanvasScaler* only works for the elements that are inside the *Canvas*. All other game objects uses the *ScaleGameObjects* script so they can be scaled automatically. This script calculates the width and height of the game object depending on the resolution of the screen and taking into account the reference resolution that will have the same value as in the *CanvasScaler* (1920x1080).

### 3.1.6 Saving and loading system

Being didactic mini-games that are made at the moment and you get a result, it is not necessary to save a lot of information. At the end of a mini-game, it always go back to

Figure 9: *CanvasScaler* component window of Unity

the main game scene where it have to identify the three digits again. So it is necessary to save some values to know the next level that the player unlocked. The Unity *PlayerPrefs* feature allows to save and access values. Only *integers*, *floats* and *strings* can be saved. To handle the change of scenes are used buttons. For example, if I finish the first mini-game, it update the variable (`PlayerPrefs.SetInt("level",2)`, `PlayerPrefs.Save()`) so that when the player returns to the main game scene this variable is loaded (`PlayerPrefs.GetInt("level")`) and the second mini-game button appears. In addition, the player can return to the menu and continue the last level he has unlocked, but if he closes the game it will restart the levels and he will begin at first mini-game. When having a global score of success and failures will always remain saved even if the player leaves the game. It can be restarted if the player uses the restart button in the initial menu.

### 3.1.7 Particles System

The *Particle System* (see Figure 10) that Unity offers to create different effects has been used. This component allows to modify the parameters of the particle to adapt it to your preferences. The parameters that I have most used for my effects are the emission, the figure, the lights, the speed and the color for the time of life, the render and also use textures. This system of particles helped me to realize the effects of appearance and disappearance of the toys, the floating particles in the air at the main game scene and also for the appearing toys. It is very useful to give visual appeal to the scene, apart from the fact that it can provide information to the player if the mathematical exercise has been successful or failed.

### 3.1.8 Audio Source

To play the audio in the videogame, the *Audio Source* component of Unity has been used. Allows you to play audios individually, pause or stop them. You can also adjust the audio volume at any time and prioritize them. As well, you can play several audios at the same time or leave the same audio in a loop. To control the audios I have the *MusicScript*, a class that contains public functions that reproduce each one of the sounds and effects. In this way I can call these functions from any class, so that I can play an audio or a sound effect in particular. It is very comfortable to use, also provides different filters for the audios.

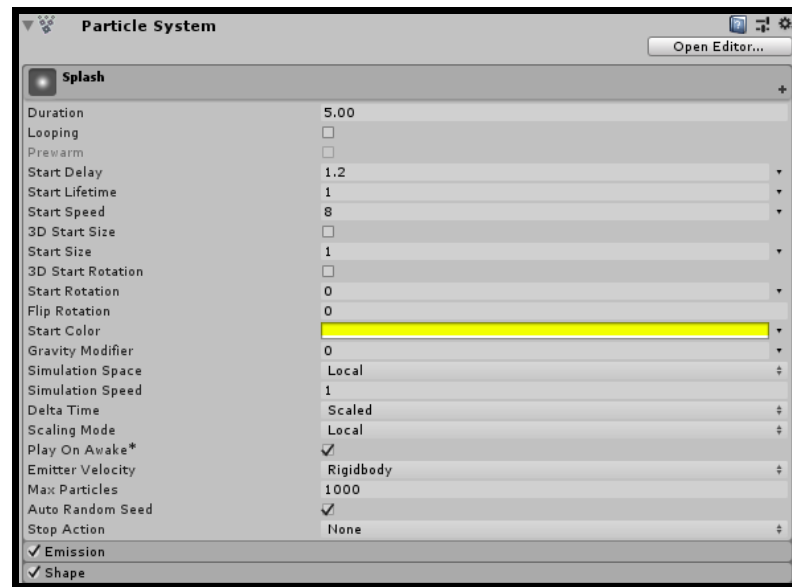


Figure 10: *Particles System* component window of Unity

## 3.2 2D Art and 3D Modeling

### 3.2.1 Menu

For the menu a background has been drawn with the title of the videogame in which is presented to the player's travel companion, Matitas (see Figure 11).



Figure 11: Background of the menu scene

### 3.2.2 3D Game Scene

Different low poly 3D models have been designed for the environments. These environments are formed by different types of trees, rocks, earth cubes, water, grass and decorative elements. Besides, materials have been applied to each of the models. At the end the models all have been put together in a single model to create the environment (see Figure 12) and facilitate the incorporation into Unity. In total there are 3 different environments in which different color palettes have been assigned to each level, with this aim to achieve the feeling of progression to the player.

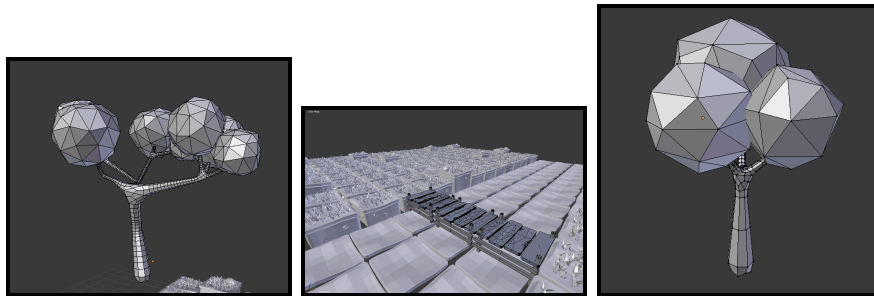


Figure 12: Models of the environment and trees

The first environment (see Figure 13) has a cold color palette, reminiscent of winter, creating the feeling that the player is far from the goal. The second environment (see Figure 14) has a warmer palette, with similar colors: red, orange and yellow, typical of autumn. The third environment (see Figure 15) has a palette that acquires tones that contrast chromatically with each other. Own shades of spring, and that being more unreal, more consistent with the fantastic theme of a world of toys. Finally, the digits have been modeled based on a recognizable and childlike text source. They also have a strong color to be visible in all environments.

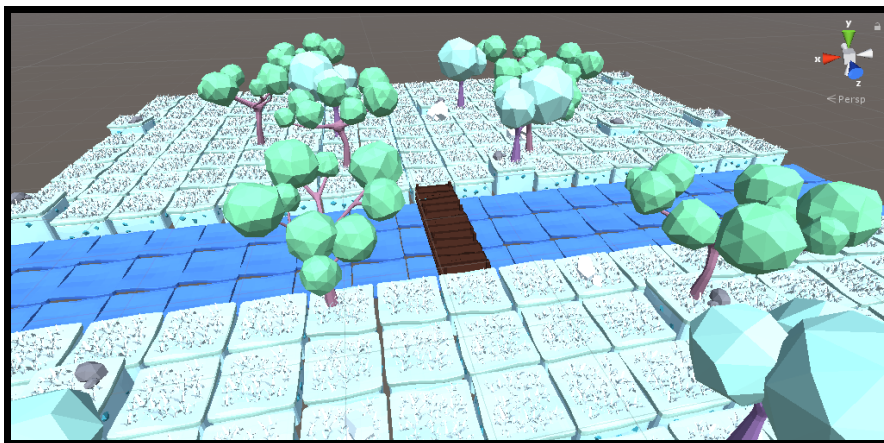


Figure 13: First final environment with materials



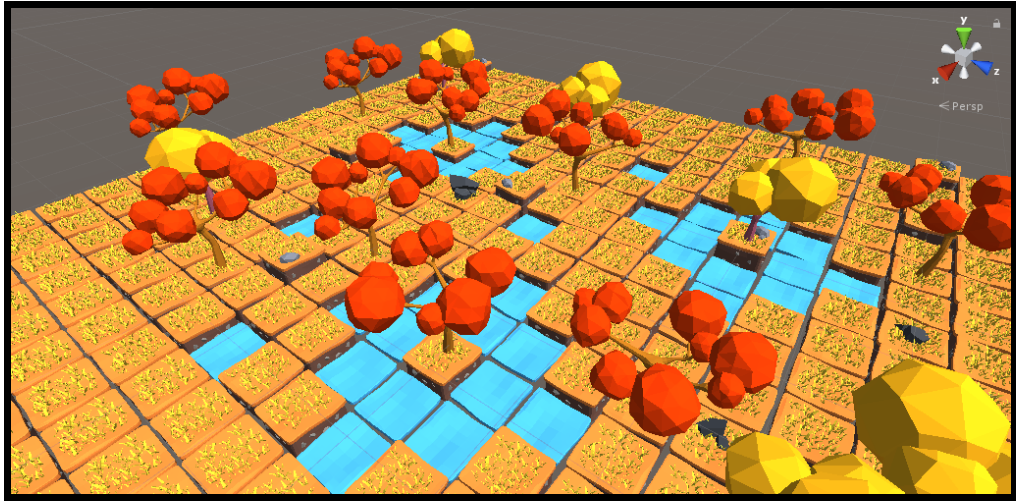


Figure 14: Second final environment with materials

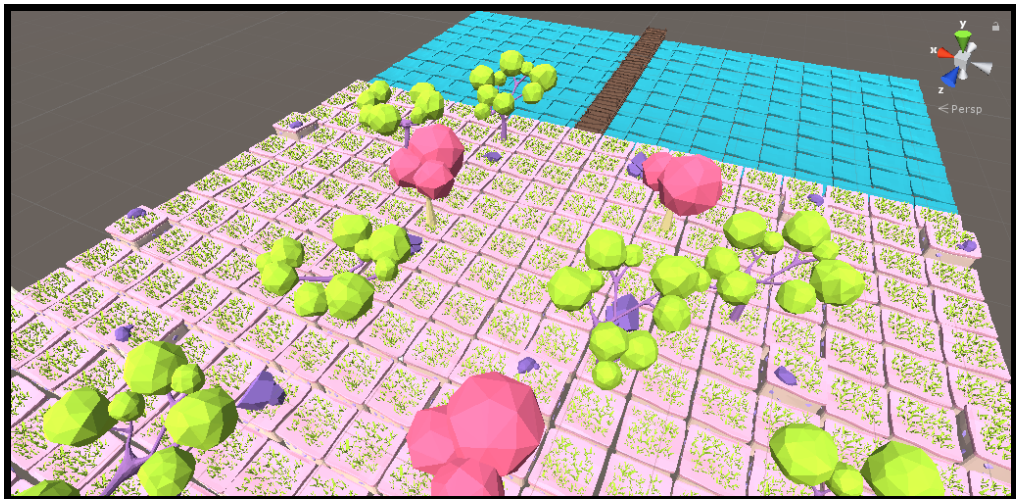


Figure 15: Third final environment with materials

### 3.2.3 Mini-games

The 2D backgrounds drawn match the 3D environments to create a relationship with each other and make sense of the world in which the player is located. There are 3 backgrounds one for each season of the year: winter (see Figure 16), autumn (see Figure 17) and spring (see Figure 18).



Figure 16: Background for the first mini-game



Figure 17: Background for the second mini-game



Figure 18: Background for the third mini-game

### 3.2.4 Character

The final result of Matitas with the three states (see Figure 19). It is green to transmit tranquility, good luck and hope.



Figure 19: 2D Sprites of Matitas and his states

### 3.2.5 2D Toys

In total I have drawn 10 toys (see Figure 20) that represent childhood and youth, with bold colors and adorable shapes to achieve the appropriate appeal. These are the toys that the player has to count.

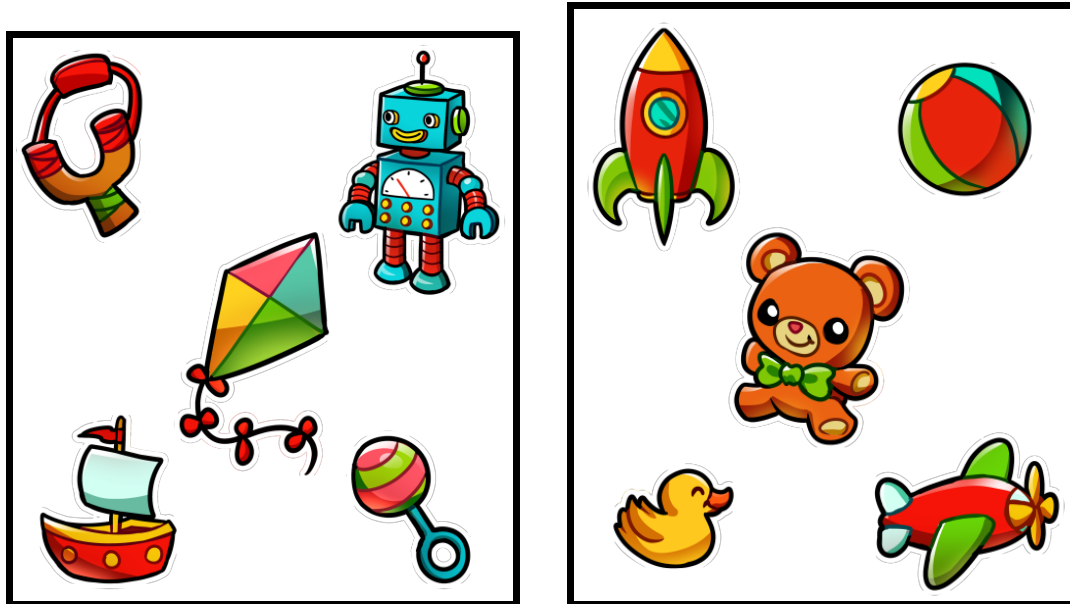


Figure 20: 2D Sprites of all the toys

### 3.2.6 UI Elements

#### 3.2.6.1 Buttons

The sprites of the menu, help, check and reset buttons have been taken from the Unity Asset Store [11] for free (see Figure 21). In addition, a rectangular platform has been drawn for all the buttons of the videogame and the dialogue (see Figure 22). Besides, colors have been used that match all the backgrounds and environments to achieve the overall relationship of the scenes.



Figure 21: Menu icons

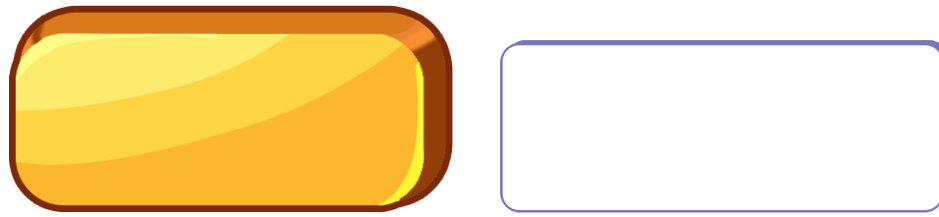


Figure 22: Buttons and the dialogue platform

### 3.2.6.2 Fonts

The text font used throughout the game has been purchased from the DaFont [12] website for free. I used two fonts, one for the whole text called BalooBhai-Regular and the other for the text in game called Coiny-Regular.

### 3.2.6.3 Icon

The icon is created from the initial title of the videogame, Dystoy (see Figure 23).



Figure 23: Icons for the executable videogame

### 3.2.6.4 Textures for particles

For the texture of the particles I used simple geometric figures to give it a more mathematical and different ambience (see Figure 24).



Figure 24: Texture for the *Particles System*



### 3.3 Music

Music is a very essential element of the videogame. It contributes a great amount of information to the player to know what is happening around him. The music of the videogame is composed of 2 background music and 3 sound effects. Taking into account the genre of the videogame and the audience has to have a slow and childish rhythm, that is catchy and at the same time fun. The first background music is for the main menu scene and the second for the main game scene and all the levels. The first is to give a cheerful and catchy atmosphere, while the second is to convey relaxation and comfort, that the player does not feel stress or anxiety while is doing the exercises. Finally, the sound effects help to understand the events that occur in the environment. This way it can know what are doing or what have to do. The sound effects are in the buttons and also in events that occur (answer correctly or wrong). Both sound effects and background music come from websites dedicated to the publication of audios for free. The background musics are from FreeSfx [13] and OpenGameArt [14]. Also, the sound effects are from FreeSound [15].

### 3.4 2D/3D Animations

Without animation, the videogame would lose its fluidity. It needs to have animations to try to simulate reality or approach as much as possible. This is an essential point to give life to the environment and that the young audience feels more attracted. For the development of animations, the *Animator* provided by Unity has been used, it is a controller that allows you to build the hierarchy of animations with states. Also change the speed between the transitions of the animations and the duration. Besides from the scripts can control which animation will run at a certain time. The Unity *Animation* system has also been used for changing positions, scaling, rotations and color changes. Using the timeline and keyframes, properties that can be modified are added and a clip for the state is created from these modifications. In the main game scene a levitation animation has been used for the digits and for the mini-games many elements use the change of position, scaling, changes of color, transparency or several simultaneous modifications. Several clips animation have been used for each of the toys, to give it a more fantastic and unreal effect.

CHAPTER

4

# RESULTS

## Contents

4.1	Work development . . . . .	<b>28</b>
4.1.1	Project deviations . . . . .	29
4.1.1.1	Temporary deviations . . . . .	30
4.2	Results . . . . .	<b>30</b>

## 4.1 Work development

In this section, all the relevant work developed and the objectives that have been reached are discussed. On the other hand will explain the deviations that the project has had and also the temporary deviations.

For the development of the game, the technical document plan (Table 1.1) that will be developed below has been followed.

In the first place, an investigation on dyscalculia has been carried out to understand what this disorder entails and what are the best ways to help the person. It is essential to put yourself on his skin and see the difficulties that are presented. Subsequently the Game Design Document (Chapter 2) has been written. This document includes all the information from which the project will start for the development of the videogame. Next, the programming started with the videogame menu. At this point the tasks have been split linearly since the mini-games are independent of the main mechanics. First the Main Game, the mini-game 1, the mini-game 2 and finally the mini-game 3.

Before starting with the art of videogames, all the mechanics and levels have been tested to correct all the possible failures. Therefore it has been followed with art. In this part it have drawn a home screen, a background for each level, the character, toys, UI elements and the icon. Having everything prepared is more convenient since the only thing that is missing is to integrate the art, music and effects. After has been done small tweaks and the implementation of some details. Finally the final report of the project has been written.

	Tasks	Estimated hours	Hours used
<b>1. Documentation</b>	1.1 Technical proposal	5	5
	1.2 Game design document	10	10
	1.3 Prepare the final memory	50	55
	1.4 Prepare the presentation	10	10
<b>2. Analysis</b>	2.1 Research on dyscalculia	20	20
<b>3. Design</b>	3.1 Game design	20	10
	3.2 Art design	50	70
<b>4. Development</b>	4.1 Programming the videogame	100	120
	4.1.1 Menu	15	15
	4.1.2 In Game	25	30
	4.1.3 Mini-game 1	20	25
	4.1.4 Mini-game 2	20	25
	4.1.5 Mini-game 3	20	25
	4.2 Retouching and detail implementation	25	15
<b>5. Testing</b>	5.1 Final testing	10	20
<b>Total:</b>		<b>300</b>	<b>335</b>

Table 4.1: Hours of work done compared to the estimated hours

#### 4.1.1 Project deviations

In this subsection all the deviations of the project as well as temporary ones will be exposed. To begin with, as mentioned before, the project has started to be developed taking into account the Game Design Document (Chapter 2) and in some cases it has had certain changes.

First, the mechanics of the main game scene have been modified. In the beginning it consisted of memorizing 3 digits and each time the player success 1 digit, one level is unlocked. When thinking about the issue, I concluded that is difficult for a boy or girl with this disorder to memorize numbers, since they have complications in identifying the digits. Then the mechanics have been modified so that each digit is written with letters and the user has to read and identify the digit. The player has to identify 3 digits each time to unlock a level.

After the levels 2 and 3 have undergone modifications. Initially, the level 2 had to calculate the 3 operations to know the result and determine which number is smaller or bigger. Now level 2 simply has to choose directly from the 4 numbers which one is the largest or smallest of all. In the level 3 initially had to solve the operations and then



select the result with the number of toys on the screen to match. At this moment level 3 consists in solving a single operation, and only if it fails the result appears in toys to force the player to count them in case of not knowing how to solve the operation.

Finally, a score has been included so that the player can see his progress. At first, I decided not to put scores so that the user does not feel anxious about the fact that digits appeared, but after all it is good to have a personal progress so that the player is motivated and try to improve himself more and more. So little by little it will progress, and the fear of error will be removed. As for the changes in the mechanics, the level 2 and level 3 is due to the difficulty that had in the beginning. The more information got about the dyscalculia had it more clear that it is necessary to lower the difficulty and start at a lower level. Since each person can affect it to a greater or lesser extent everything depends on the type of dyscalculia. Also each one has a different way of understanding mathematics. It is better to start from the bottom and go to the top.

#### 4.1.1.1 Temporary deviations

For the development of the project there have been certain temporary deviations. In total there are 300 hours of work, but some tasks have taken longer than planned. Since certain programming tasks that I have underestimated have cost me more. In addition to the testing, I have had to correct some errors that I have also taken. On the other hand, it took me more time than expected in the art of the game because I wanted to polish all the drawings as much as possible. Attractiveness is essential to achieve my proposed objectives. The estimated hours compared to the hours used will be shown below (Table 4.1).

## 4.2 Results

At the end of the project is important to see what objectives have been met according to the objectives established at the beginning (Section 1.3).

The main objectives:

- Achieve a colorful and childish art, that is attractive and pleasant for a young audience.

This objective has been fulfilled. The art design is childish, very colorful, attractive and with a fun purpose. Also the style of art approaches the style originally proposed (Section 2.7).

- Adapt the traditional mathematical exercises based on a specific symptom and turn it into a functional mini-game that helps in the learning of mathematics in that particular aspect.

The mini-games are based on the treatment of ideognostic and practognostic

dyscalculia. Each of these types of dyscalculia covers a large number of symptoms that can be similar or totally different, everything depends on the person and age. Since the objective audience is between 6 and 9 years old, I focused on the most effective exercises. The identification of the digits, learn to count, distinguish quantities and finally perform basic calculation operations. Therefore, the objective has been satisfactorily fulfilled.

- Present a functional demo of the videogame with three mini-games.

The videogame presents a main mechanic from which the 3 functional mini-games are unlocked. The mini-games have been tested and corrected in case of error. So the goal has been successful.

The secondary objective:

- Getting the player to feel identified with the main character.

Being a secondary objective does not have much priority, but even so with the design of the art and part of the dialogues it has been tried that the player feels identified. This objective is not completely achieved since it would be necessary to develop more the narrative of the videogame.

The figures from 25 to 30 show captures corresponding to different moments of the game. In particular, figure 25 shows the initial menu of the game, the 26 and 27 the main game scene at different times of progress, the 28 the first minigame, 29 the second minigame and 30 the third minigame.



Figure 25: Initial menu of Dystoy



Figure 26: Main Game Scene before first mini-game

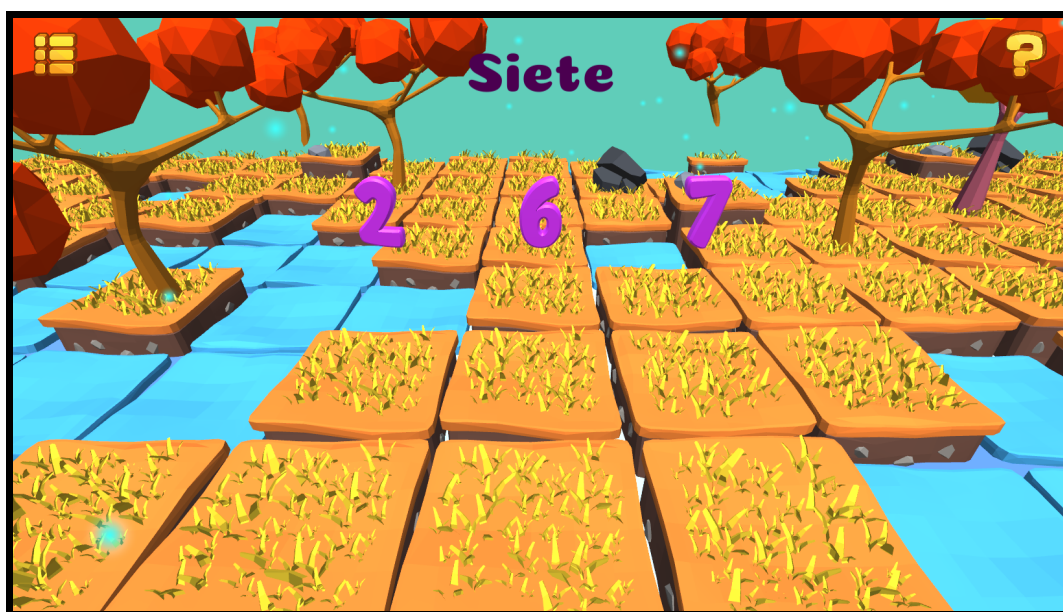


Figure 27: Main Game Scene before second mini-game



Figure 28: First mini-game scene



Figure 29: Second mini-game scene



Figure 30: Third mini-game scene

When taking into account all the main objectives fulfilled, it is concluded that the result of the project is satisfactory. A functional 2D/3D videogame of educational genre with a main mechanic and 3 mini-games focused on learning and improving math skills for people with dyscalculia has been obtained, with a visual and auditory aesthetic suitable for the objective audience. In addition, the videogame is available for the computer and the android tablet.

Below are the different links in relation to the project:

- Downloadable project:  
<https://drive.google.com/file/d/1oZxhH6h7tMb6cTEnz9zgZhTAm0b-jhNJ/view?usp=sharing>
- Build for the computer:  
[https://drive.google.com/file/d/1AFlfH6rVE\\_\\_7aSXbP43uZUA4BghH7oyq/view?usp=sharing](https://drive.google.com/file/d/1AFlfH6rVE__7aSXbP43uZUA4BghH7oyq/view?usp=sharing)
- Build for the tablet (Apk):  
[https://drive.google.com/drive/folders/1xf7QTQA3J0xBaA\\_Dc65-S8cZ\\_-QPsfjq?usp=sharing](https://drive.google.com/drive/folders/1xf7QTQA3J0xBaA_Dc65-S8cZ_-QPsfjq?usp=sharing)
- Video that shows the gameplay of Dystoy:  
<https://drive.google.com/file/d/1lp0Wi90kbBYsALhYRkwmNRDCyJ-rjlid/view?usp=sharing>



# CHAPTER 5

## CONCLUSION

### Contents

5.1	Conclusions	35
5.2	Future work	36

### 5.1 Conclusions

As a result of the above, I am very happy with the project. I have put to the test all the knowledge obtained throughout these years. I have also learned much more about some aspects of videogames with an educational objective or serious games. At first, I have underestimated this genre of videogames and is much more complicated than it seems at simple view. Designing mechanics that can help and educate a target audience with dyscalculia or simply for the education of people with or without difficulties is difficult. You have to investigate in depth the problem and in what way I can adapt the difficulties it suffer to help them.

On one hand, the difficulty is another issue. Each person has different characteristics and personalities and will respond in one way or another to a specific problem. You also have to take into account where you want to go and in what way I can use the videogame as a tool to achieve knowledge. It has to be understandable, concise, motivating and above all entertaining. As well, it is important to create emotions and to have a pleasant experience, to learn. In my opinion, videogames as an educational tool will be the future in a few years, since it has a very positive effect, reduces costs and is very accessible.

## 5.2 Future work

In the future I would like to continue developing the game, at this moment it is a demo. I have several ideas in mind that due to lack of time are not implemented. First of all, developing a more intelligent AI algorithm that can detect which digits the player has more difficulty at identifying and base the exercises on that digit. Also, save the operations or calculations that he does not understand, and if he confuses the arithmetic symbols.

Secondly, add more amount of mini-games and that the player himself can select the one he prefers. An example of a minigame is that the player has to hear a digit by audio and has to select it, that is able to relate them, or a large number of toys appear and it have to count the toys that are indicated. Third, a system for difficulty. Before starting the game the player is presented with a brief test to make a calculation of all their faculties and determine the difficulty of the videogame. Also, you can change the difficulty yourself.

If the time comes when the game is complete, I would like to upload it to Google Play for free. I have always wanted to make a videogame that helps people because it is rewarding and motivating to see how creates emotions in the person who is playing it.

## BIBLIOGRAPHY

- [1] A. Morin, “Tratamientos para niños con discalculia.” <https://www.understood.org/es-mx/learning-attention-issues/treatments-approaches/treatment-options/treatment-options-for-dyscalculia>. Accessed: 2019-04-17.
- [2] A. M. Lee and J.D., “How games can help kids who struggle with math.” <https://www.understood.org/en/school-learning/learning-at-home/games-skillbuilders/how-games-can-help-kids-who-struggle-with-math>. Accessed: 2019-04-17.
- [3] Understood, “Experience it: Math issues.” <https://www.understood.org/en/tools/through-your-childs-eyes/player?simq=d7e339d7-7c55-4fd0-bef6-94b1a640d9dc&standalone=true&simulation=true>. Accessed: 2019-04-19.
- [4] K. health GmbH, “Meister cody talasia.” <https://www.meistercody.com/educationalresearch/talasia-math/?lang=en>. Accessed: 2019-04-19.
- [5] A. Lussier, “Blips!” <https://cenop.ca/cenop/application/blips-les-robots-de-la-numeration/>. Accessed: 2019-04-19.
- [6] Wikipedia, “Dyscalculia.” <https://es.wikipedia.org/wiki/Discalculia>. Accessed: 2019-04-17.
- [7] Fandom, “Navi, the legend of zelda.” <https://zelda.fandom.com/es/wiki/Navi>. Accessed: 2019-04-17.
- [8] Ubisoft, “Rayman adventures.” <https://www.ubisoft.com/es-es/game/rayman-adventures/>. Accessed: 2019-04-17.
- [9] Mothman64, “Pancakes & milkshakes.” <https://www.deviantart.com/mothman64/art/Pancakes-and-Milkshakes-Fan-Made-Game-740108145>. Accessed: 2019-04-17.
- [10] Disney, “Wreck-it ralph 2.” [https://es.wikipedia.org/wiki/Ralph\\_Breaks\\_the\\_Internet](https://es.wikipedia.org/wiki/Ralph_Breaks_the_Internet). Accessed: 2019-04-17.
- [11] U. A. Store. <https://assetstore.unity.com/>. Accessed: 2019-05-12.
- [12] DaFont. <https://www.dafont.com/es/>. Accessed: 2019-05-12.
- [13] FreeSfx. <https://www.freesfx.co.uk/>. Accessed: 2019-05-12.
- [14] OpenGameArt. <https://opengameart.org/>. Accessed: 2019-05-12.
- [15] FreeSound. <https://freesound.org/>. Accessed: 2019-05-12.